加微信：642945106 发送“ 赠送 ”领取赠送精品课程
发数字“ 2 ”获取众筹列表

06 | 接口测试平台：工具和框架不可以兼容？

2020-02-12 陈磊

接口测试实战课 进入课程 >

讲述：陈磊
时长 11:29 大小 9.21M

你好，我是陈磊。很高兴在接口测试课程中再次遇见你。

到目前为止，我们的课程重点介绍了完成测试任务的两种接口测试手段，第一种是使用如 Postman 这样的工具，第二种是打造属于你自己的测试框架。上节课我们还一起学习了 RESTful 风格的接口，并针对它的特点完善了我们自己的测试框架。

这节课我就教你如何用工具和框架的组合搭建接口测试平台，让你能更快速地完成测试任务。

工具的便捷性与框架的灵活性可以兼得

说到这儿，你一定有一个困惑，在前面我先讲了 Postman 这款非常好用的 HTTP 测试工具，后来又讲了怎么自己动手封装接口测试框架，它们各有特点，比如工具有便捷性，框架有灵活性，这无疑是两条都可以通向罗马的路，是两种都可以完成接口测试工作的方法，那学会一个不就可以了，为什么两个都要学会呢？

而且工具和框架，这两件事看起来互不相干，甚至有些互相排斥，那么这两种接口测试技术手段能相互支持，能融合到一起吗？下面我就来回答这个问题。

其实，工具和框架，这两条通向罗马的路可以并成一条快速通道，让你大踏步进军罗马。所以我既建议你要掌握一款好用的工具比如 Postman，也建议你用自己的技术沉淀出自己的框架，如果你能正确地混合使用它们，实质上就可以搭建起一个接口测试平台，帮你更快速地完成测试任务。

在脚本的设计过程中，这样做有两个好处。

一是能充分发挥 Postman 界面化的优势，快速完成大量的脚本撰写工作；二是通过你自己的框架完成测试脚本的执行，所有的过程代码都会存储到你自己的代码仓，这样，既可以留下测试的过程资产，也便于版本控制，这也为持续集成、持续交付等平台提供了无人值守的、按需驱动测试的途径。

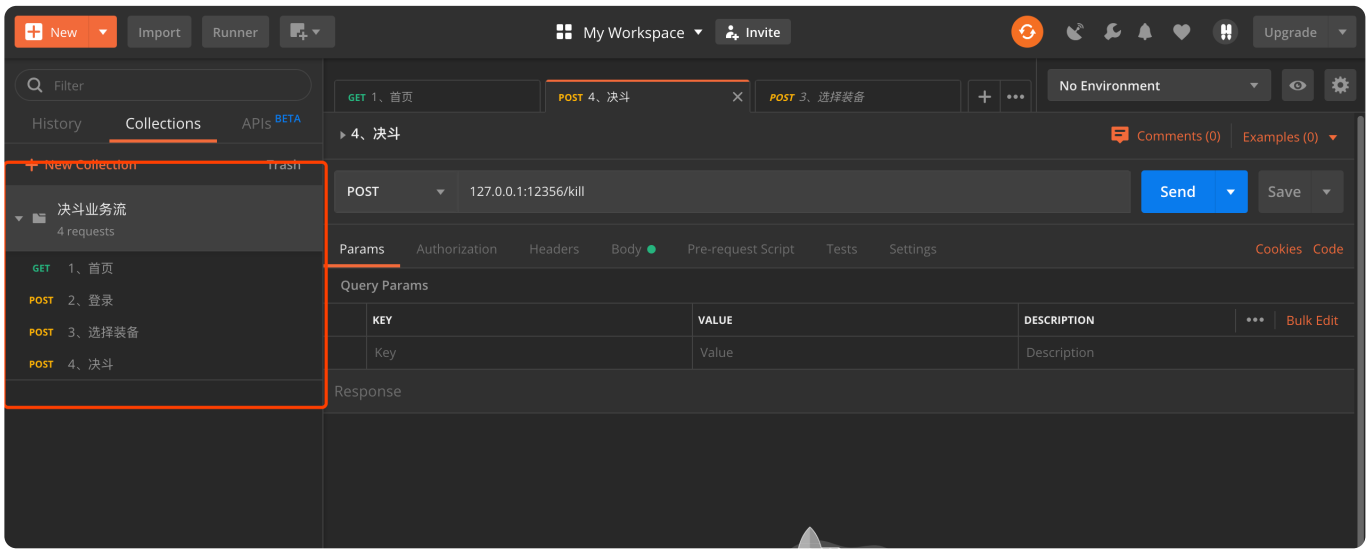
此外，这样做也可以提升团队的工作效率。

在我的团队中，有一些小伙伴很不喜欢写代码，相反，他们更喜欢使用工具，工具用得也非常溜，我相信在你的团队中也会存在这样的情况。但是，仅仅依靠工具，只能一个人完成一件事情，这并不方便团队内部的团队合作、交接和技术积累。

但是，我们又没有办法让所有人一下子都喜欢上写代码，那么该如何降低代码编写门槛呢？通过工具和框架搭建接口测试平台，其实就是一个很好的解决方案。这样，你既可以让你的团队有技术积累，又能给团队中一些编码能力比较薄弱的小伙伴学习时间，最重要的一点是，这不会影响整个工作的进度。

工具的便捷性可得

不知道在学完前面的课程后，你是不是还用过自己的 Postman，当你再次打开 Postman 的时候会发现，你之前用它来完成的测试脚本是被保存下来了的，就如下图所示：



可以看到，我上次使用 Postman 测试“战场”系统的脚本还是存在的，如果你忘记了，可以在课后看一看你用来做测试的 Postman，再分别看看那四个请求。

到这里，我想请你先闭上眼睛回想一下，你之前使用 Postman 做接口测试的整个流程，是不是清晰可见？你也可以同时回忆一下，你用自己封装的 Common 类，编写“战场”系统测试脚本的过程。你会发现，和工具的使用过程相比，在这里你不太容易回想出自己每一步都做了什么。

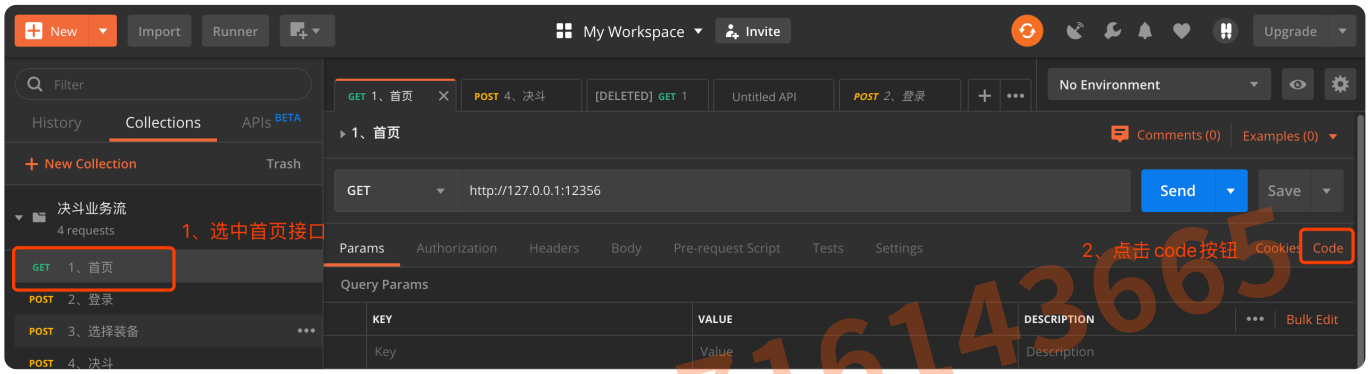
这也是 UI 操作和代码操作的区别之一，UI 操作更加直观，可以在你的脑海里留下更深刻的印象；而代码操作给人留下的印象就比较模糊，但是，通过用代码写脚本来完成接口测试，比较便于维护、团队合作以及留存。

讲到这你肯定会问：“你把用 Postman 类工具完成接口测试，以及自我封装测试框架这两种方法各打五十大板，那它们到底哪个好？”其实我的目的并不是想让你分出个好坏，好坏

之分都是相对的，每个人的习惯和喜好都不相同，但是我们却可以把它们的优点都利用好，把这两种技术的优势都发挥出来。

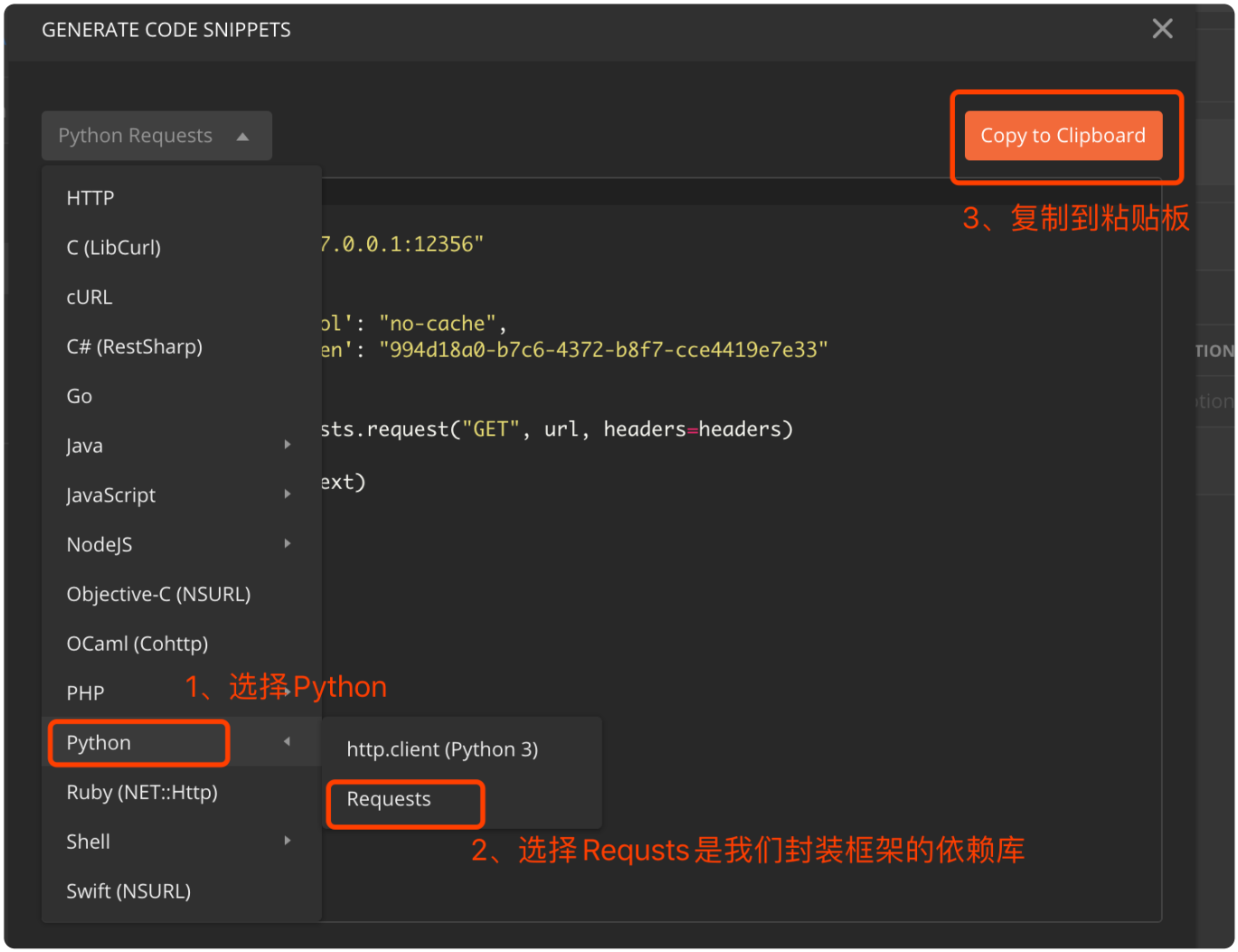
我们利用 Postman 设计接口测试直观、快速的优势，将它变成接口测试脚本的初始脚本的编写工具，其实 Postman 也可以配置 Chrome 插件录制请求，这些在 Postman 官方已经有很详细的介绍，所以我就不在这里详细讲解了，如果你感兴趣，课后可以自行学习。

我们以之前的测试脚本为例，选择第一个单接口接口测试的脚本，在右侧点击 Code 按钮。



在弹出框中，你可以选择各式各样技术栈的测试脚本，在这里，我们还是用在之前例子中所选取的 Python，我们框架的依赖库是 Requests，这样你就可以看到显示出的代码了，就如下图所示。看到这些代码，你是不是已经开始觉得，通过这样的处理来编写脚本更加容易。

拼课微信：17161436655



由此可见，和写代码相比，使用 Postman 来设计接口测试要更容易使用，对于代码基础比较薄弱的测试工程师来说，这种方法也更容易掌握。

框架的灵活性亦可得

在刚刚高兴的心情慢慢冷静下来以后，你是不是在心里默默地埋怨我？既然有这么简单的方法，为什么还一直让你学习一门编码技术，还建议你如果什么都不会，可以学习一下 Python？这是因为工具生成的代码可读性特别差，也并不适合我们将它作为团队的技术积累留存。

现在，我们一起看一看由工具生成的代码。先来看看第一个接口首页单接口对应的代码：

复制代码

```
import requests


url = "http://127.0.0.1:12356"

```

```
headers = {
    'cache-control': "no-cache",
    'Postman-Token': "8c6247bb-744a-43d3-b27d-9e51af923c5d"
    }


response = requests.request("GET", url, headers=headers)


print(response.text)

```

上面的这个代码你是不是似曾相识？这就和我们第一次写的第一个接口的单接口测试代码一样，是一个流水账一样的脚本，这些代码如果原模原样地存到你的代码仓中，对你再次使用它没什么好处。那么在这基础上，我们可以将它修改成自己框架的脚本，就如下面这段代码所示：

复制代码

```
    # 引入你的框架
    from common  import Common
    #访问uri
     uri_index = "/"
     #调用你的Common类
     comm = Common('http://127.0.0.1:12356')
     # 完成方法
     response_login = comm.get(uri_index)
     # 打印response结果
     print('Response内容: ' + response_login.text)
```

这个代码你是不是很亲切？Common 类可是我们的老朋友了。那么接下来，我们再看看第二个接口登录的单接口测试脚本，你可以用相同的方法，找到它的 Python 代码，为了方便有些不是很方便打开自己 Postman 的同学，我把对应的代码放到了下面：

复制代码

```
import requests


url = "http://127.0.0.1:12356/login"


payload = "username=criss&password=criss"


```

```
headers = {

'cache-control': "no-cache",

'Postman-Token': "fdc805e1-4406-4191-ae44-ab002e475e03"

}

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

如果你还记得我在测试代码及框架那一节课（也就是04）中讲的内容，就会发现，它和那里最开始部分的代码实现几乎一致，这和我们自己手动写的代码最大区别就是，它少了很多注释，而多出一些访问头信息，也就是上面代码的 headers。

headers 在我们“战场”系统的测试中并不是必须传递的参数，但是 Postman 这种工具会将其添加默认值传递给服务器。这是由这个工具添加的，你在写脚本时，如果它是非必填的，你可以忽略它。但是，工具这么做是为了匹配所有的情况，所以它会做一些和我们这次测试不相干的工作。

难道说 Postman 这么好的功能，对我们来说就没有一点好处吗？其实我们在上面代码的基础上，将其修改成引入我们自己框架的测试代码，完成修改后，再推送到接口测试项目的代码仓中，就如下面这个代码所示：

复制代码

```
from common import Common
uri = "/login"
payload = "username=criss&password=criss"
comm = Common('http://127.0.0.1:12356')
response_login = comm.post(uri_login,params=payload)
print('Response内容: ' + response_login.text)
```

这也无疑加快了我们测试脚本的编写，在上面这个过程中，我们也很容易再次发现需要封装到框架中的公共方法，这样循环下来，就加快了我们测试脚本的积累速度，同时我们也就可

以有更多时间用在框架的维护上了。

通过代码的改写和封装，我们就可以将工具生成的代码完美结合到我们的框架中了，当我们需要修改已经存储在代码仓库的脚本时，我们只需 pull 代码仓的代码，就可以看到易读、易维护的测试脚本了。

总结

我们今天的课程到这里就结束了，现在你闭上眼睛回顾一下，如果你的头脑中就只有用 Postman 快速编写脚本，自己框架留存执行的话，只能说明你今天学习得很认真，但是并没接受我想告诉你的主旨思想。

我今天以 Postman 工具和你自己的框架相结合的例子，告诉你如何建立一个你自己的测试平台，你可以通过三步完成工具加框架的组合方式：

1. 借助 Postman 这类工具的易学、易操作的特点，将它变成你测试脚本中快速创建的脚本撰写工具；
2. 利用工具提供的导出代码功能，将其导出成我们流程化的测试代码；
3. 通过我们自己的框架，改写我们通过工具导出的脚本。

最后，你的测试脚本可以存入代码仓中为持续集成平台提供持续验证，这就完成了一套简单又灵活的接口测试平台的建设。

实际上，在本节课中，我更希望帮你建立一种解决问题思路，测试工程师的技术普遍会稍微弱于开发工程师，你要善于利用各种技术手段来帮助自己解决问题。

无论你团队中的小伙伴是用 Postman 生成测试脚本，再通过修改集成自己的框架，还是直接通过框架写测试脚本，它们都殊途同归，都是以最终统一的方式推送到了代码仓库中。这样，就不会让代码能力变成阻塞最终工作的一个关键节点，同时，这对于使用 Postman 编写脚本的小伙伴来说，他们也会越来越熟悉自己的框架，逐渐提升自己的技术能力，并加强自己的代码能力。

思考题

今天的课程中，我们把“战场”系统的测试脚本又拿出来测试了一次，不知道你是不是有点厌烦了，今天我仅仅给你演示了第一个和第二个单接口测试脚本，它们从工具到框架的演变过程，那么，你可以将后面的两个单接口测试脚本自己完成，并在留言区回复给我吗？如果你已经开始将这个方法应用到你的工作中了，那么我也希望你能将自己在使用过程中的心得体会分享给我。

我是陈磊，欢迎你在留言区留言分享你的观点，如果这篇文章让你有新的启发，也欢迎你把文章分享给你的朋友，我们一起来探讨和学习。

课程学习计划

关注极客时间服务号
每日学习签到

月领 25+ 极客币

【点击】保存图片，打开【微信】扫码>>>

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　05 | 案例：测试框架如何才能支持RESTful风格的接口？

下一篇　07 | WebSocket接口：如何测试一个完全陌生的协议接口？

精选留言 (13)

写留言

日拱一卒
2020-02-25

通过工具和自建测试框架相结合，提高测试case创建效率，是一个很好的思路，希望能对广大测试同学有所启发。

关于如何将测试用例导入到自建框架中，我建议中间能通过程序做一些自动化转换，我们可以定制规则，将更快速的生成测试用例，例如，将工具导出的测试用例，直接保存到d...

展开

王富贵

2020-02-24

postman能批量导出python脚本吗

展开

作者回复: 目前我还没有发现这个功能，不知道最新版本中是否支持这种批量导出，但是批量导出会不会在修改脚本的时候会产生会乱呢？我确实也没有怎么考虑过该问题，如果你有类似时间欢迎留言给我。谢谢

派森

2020-02-19

大佬，因为有时研发现场开发的接口，连个文档都没有，时间太紧张，没有精力去编写自动化用例，但是回归测试又必不可少，如果全部手工操作，测试组的压力很大，本身我们是有基于RF的Web测试框架，但是不适用这种模式，所以就想到利用接口的录制和回放，想省去编写用例和代码这部分工作，这块有没有好用稳定的框架或者工具介绍下啊

展开

作者回复: 录制回放可以使用poatman的方式，在框架中我知道httprunner支持但是具体我并没有深入研究过，您可以考察一下是不是适合您team。

1

aoe

2020-02-18

作为一个Java程序员，看了老师的这篇文章，感觉非常适合推动自动化测试，让不懂代码的测试小姐姐快速入门，提升战斗力。

展开

作者回复: 谢谢您的支持。

、Attention

2020-02-16

!/usr/bin/env python
-*- coding: utf-8 -*-

from common import Common
...

展开

作者回复: 谢谢，正则表达式是可以完成的，也可以用json库将其变成对象进行处理

hello

2020-02-14

老师，请教个问题，接口测试是否依赖特写的环境，对应环境中产生及依赖的数据该如何维护，多谢

作者回复: 您好，接口测试的依赖问题确实有问题。因此需要使用mock来解决，我会在后续课程中讲解mock 服务，并推荐一个简单容易使用的mock工具。

京都念慈菴

2020-02-14

自己开发能力基本没有，但是之前看过一些纯代码构建测试框架的教程，一直无法坚持下去，用了工具又觉得有时候很受局限，真没想过把他们并存，现在我能即开始这些工作又能同时提升自己开发能力，将他们结合在一起

展开

作者回复: 谢谢，太棒了，我希望你能在我的课程中有所得，也感谢你的留言

一叶知秋

2020-02-14

我有过一个类似的实践，将postman中所有的接口信息导出到一个名为postman.josn文件中，然后用python脚本提炼出一个接口名称、请求方式、body等等信息 然后存入数据库中，然后接口自动化时的请求主体信息从数据库获取，新建领一张测试用例表（主要是存

储payload信息的，针对输入做用例）从而可以配置化地做接口自动化，目前我还没有全部完成，老师你觉得这个方案可行吗。

展开

作者回复: 您好，您的思路也很好，是一种解决问题的方法。继续加油完成你的实践过程，期待您成功后来分享您的实践心得和技术细节。

小鱼儿

2020-02-13

自从写脚步就比较排斥工具，通过这节课学到了工具的另类作用，赞

作者回复: 谢谢您的支持，希望在专栏里面你有所得，更期望把你的想法和体会告诉我，一起进步学习。

happychap

2020-02-12

这种方式可以轻松将gui工具的成果入代码仓，与ci打通，挺不错的。不过也存在一个小缺点，莫接口发生部分变化，你得针对该接口重新撸一遍，单接口以前的转换成果就没办法复用了。
关键收获还是老师提出的思路，举一反三会有更大的收获。

展开

作者回复: 谢谢您，谢谢支持，您比我总结的更抽象，说明你也是个测试的技术老手。谢谢支持

滚球兽

2020-02-12

挡板服务开发框架一般有哪些呢？列举一下么？想做个挡板服务

作者回复: 您好，挡板系统也是我们常说的mock服务，但是更专业的交付叫 test double。我会在最后一节课介绍一个简单易用的mock框架。我比较推荐使用一个比较流行的框架完成比如 mockito、javamock等，很好用又很简单，除非必须自己研发否则现在开源的这些已经很优秀了

安明

2020-02-12

目前这篇文章只是方便了一个不会编码的人员、，利用postman工具生成脚本。例如：脚本的执行、参数化数据、response的测试脚本，这些代码，还是要在自己的框架基础上自行编写。怎么和postman工具结合呢？？

作者回复: 您说的特别好，也是一个很好的问题。这种方法很适合基础薄弱的小伙伴，其实也可以给代码大神使用。这是因为速度和质量效能才是我们不断的在推广提高的，这是一个快递完成脚本撰写的方法。谢谢您的留言

Middleware

2020-02-12

学到了，用了这么久 Postman 原来还可以这样？👍👍

展开

作者回复: 谢谢支持